Modbus
Remote Communication Protocol for REX 521

Technical Description

Industrial IT enabled products from ABB are the building blocks for greater productivity, featuring all the tools necessary for lifecycle product support in consistent electronic form.

Contents

1. Protocol overview

The Modbus protocol was first introduced by Modicon Inc. and is widely accepted as a communication standard for industrial device controllers and programmable logic controllers (PLCs). The protocol determines how each controller connected to a Modbus network will recognize a message addressed to it. It also determines the task to be performed and extracts any data or other information contained in the message. If a reply is required, the controller will construct a reply message and send it using the Modbus protocol.

A master device can be connected to slave devices either directly, or via modems using a compatible serial interface. The interface defines the connector pinouts, cabling, signal levels, transmission baud rates, and parity checking.

The communication technique used in the Modbus protocol is a master-slave technique. This means that only one device can be the master and initiate transactions while other devices connected to the network are slaves and can accordingly not initiate any transactions.

A message sent by the master to the slave is called a query; see Figure 1.-1. The master can address a query to an individual slave or to all slaves, that is, to broadcast the query. After the slave has received a query, it attempts to perform the requested task. If a query has been sent to an individual slave, the slave will send a message, that is, a response to the master. However, if it has been broadcast, no response will be sent. The response can be either a normal response (in case of performing the requested task) or an exception response (other cases).

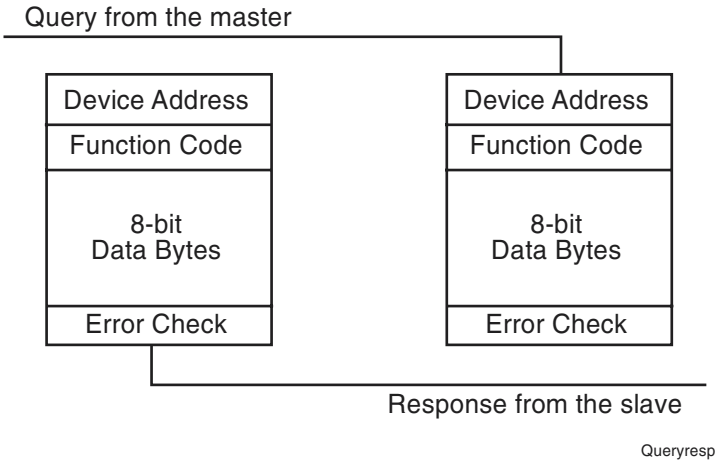

Fig. 1.-1 Query response cycle

There are four types of Modbus data: digital inputs (DI), input registers (IR), coils (CO), and holding registers (HR). Each type of data consists of either scan or control points, which all have separate 16-bit addresses.

All the data addresses in the Modbus protocol are referenced to zero. The first occurrence of a data item will be addressed as item number zero. For instance, coil 1 will consequently be addressed as coil 0, and coil 127 as coil 7E hex (126 in decimal format).

1.1. Transmission frame formats

The query format in the Modbus protocol is as follows:

- The first byte of the frame is the address of the slave to which the query is directed. If the query is broadcast, the address is 00.
- The second byte is a function code defining the requested task to be performed.
- The following bytes are the data to be sent and the last two bytes form an error-checking field.

The response includes fields containing

- the address of the slave,
- confirmation of the performed task in the form of a function code,
- any data to be returned, and
- an error-checking field.

If an error occurs in the receipt of the message, or if the slave is unable to perform the requested task, the slave will construct an error message and end it as its response.

The Modbus protocol has two serial transmission modes: ASCII and RTU. The transmission modes define the bit contents of the message fields transmitted in the network. They also determine how the information is packed into message fields and how it is decoded. The selected mode and the serial parameters must be the same for all devices in a Modbus network.

1.1.1. ASCII mode

In the ASCII (American Standard Code for Information Interchange) mode, each byte in a message is sent as two ASCII characters forming a hexadecimal number. The allowable characters are hexadecimals 0-9 and A-F.

The advantage of the ASCII mode compared to the RTU mode is that it allows time intervals of up to one second to occur between the characters without causing an error. The character format is presented in Table 1.1.1-1 on page 5 and the character bit sequence in Figure 1.1.1.-1.

Table 1.1.1-1 ASCII mode character format

Coding system	Byte sent as two ASCII characters representing a hexadecimal number
Bits per character	1 start bit 7 data bits, the least significant bit is sent first 1 bit for even/odd parity; no bit if parity is not used 1 stop bit if parity is used; 2 stop bits if parity is not used
Error check field	Longitudinal Redundancy Check (LRC)

With parity checking

START	1	2	3	4	5	6	7	PARITY	STOP

Without parity checking

START	1	2	3	4	5	6	7	STOP	STOP

Fig. 1.1.1.-1 ASCII bit sequence description

The message starts with a colon character (:) and ends with a “carriage return – line feed” (CRLF) pair, that is, ASCII 0D and 0A in hexadecimal format. The other fields in the message frame are identical with those in the RTU mode with the exception of the error-checking field: Longitudinal Redundancy Check (LRC) method is applied in the ASCII mode and Cyclical Redundancy Check (CRC) in the RTU mode.

When a device which has been connected to the network detects a colon character, it will decode the following field to find out whether it is the device to which the query is directed. Therefore, each device must continuously monitor the Modbus network.

If an interval longer than one second occurs between characters, the receiving device will assume that an error has occurred. Consequently, it will clear the receive buffer and start waiting for the colon character. A typical message frame is shown in Figure 1.1.1.-2.

Start	Address	Function	Data	LRC check	End
1 char	2 chars	2 chars	2*n chars	2 chars	2 chars (CRLF)

Fig. 1.1.1.-2 ASCII standard frame format description

1.1.2. RTU mode

In the RTU (Remote Terminal Unit) mode each message character is sent in binary format. Each character has:

- one start bit,
- eight data bits,
- one even, odd or no parity bit, and
- one or two stop bits.

The number of stop bits depends on whether a parity bit is used. If odd or even parity is used, the character will have one stop bit. If parity is not used, however, there will be two stop bits. In total there are eleven bits in one character. The RTU character format is presented in Table 1.1.2-1 on page 7.

Table 1.1.2-1 RTU character format

Coding system	8-bit binary
Bits per character	1 start bit 8 data bits, the least significant bit is sent first 1 bit for even/odd parity; no bit if parity is not used 1 stop bit if parity is used; 2 stop bits if parity is not used
Error check field	Cyclical Redundancy Check (CRC)

The messages are transmitted in the network from left to right, that is, the least significant bit (LSB) first and the most significant bit (MSB) last. The RTU bit sequence is shown in Figure 1.1.2.-1.

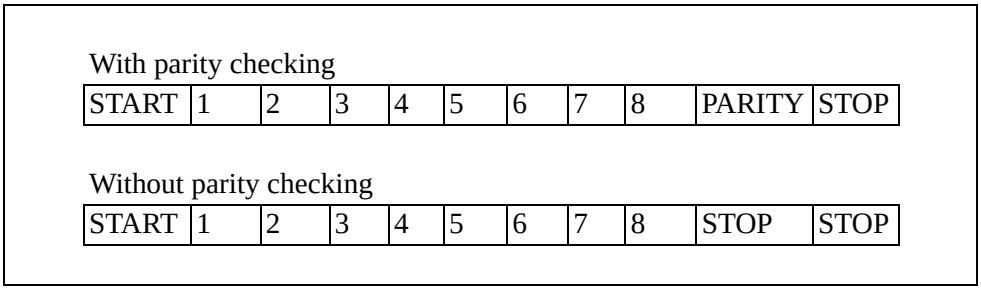

Fig. 1.1.2.-1 RTU bit sequence description

The beginning of each frame is marked with a silent interval of at least 3.5 character times. This is implemented as a multiple of character times at the selected baud rate. The end of the frame is also marked with a silent interval of at least 3.5 character times. A typical message frame is shown in Figure 1.1.2.-2.

Start	**Address**	**Function**	**Data**	**CRC check**	**End**
T1-T2-T3-T4	8 bits	8 bits	N x 8 bits	16 bits	T1-T2-T3-T4

Fig. 1.1.2.-2 RTU standard frame format description

1.2. Master's queries

The format of a master’s query depends on the function used. The format of a read function query (read coil status, read input status, read input registers, and read holding registers) is as follows:

Mode	**ASCII**	**RTU**
Start character	:	3.5 character times of the silent line
Address	2 characters	1 byte
Function	2 characters	1 byte
Starting data address	4 characters	2 bytes
Quantity of points	4 characters	2 bytes
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

The format of a force single coil or a preset single register function query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character times of the silent line
Address	2 characters	1 byte
Function	2 characters	1 byte
Data address	4 characters	2 bytes
Data value	4 characters	2 bytes
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character time of the silent line

The format of a force multiple coil or a preset multiple registers function query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character times of the silent (idle) line
Address	2 characters	1 byte
Function	2 characters	1 byte
Data address	4 characters	2 bytes
Quantity of points	4 characters	2 bytes
Byte count	2 characters	1 byte
Data values	2 x N characters	N bytes [a]
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

a. N is equal to byte count

The format of a read/write multiple registers function query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character times of the silent (idle) line
Address	2 characters	1 byte
Function	2 characters	1 byte
Data address	4 characters	2 bytes
Quantity of points	4 characters	2 bytes
Byte count	2 characters	1 byte
Data values	2 x N characters	N bytes [a]
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

a. N is equal to byte count

The format of a diagnostics function query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character times of the silent line
Address	2 characters	1 byte
Function	2 characters	1 byte
Subfunction	4 characters	2 bytes
Data field	4 characters	2 bytes
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

1.3. Normal responses

The format of a normal response to a master's query depends on the function used. The format of a response to a read function query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character times of the silent line
Address	2 characters (echo of master's query)	1 byte (echo of master's query)
Function	2 characters (echo of master's query)	1 byte (echo of master's query)
Byte count	2 characters	1 byte
Data values	2 x N characters	N bytes[a]
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

a. N is equal to byte count

The format of a response to a force single coil or a preset single register function query is an echo of the query itself:.

Mode	ASCII	RTU
Start character	:	3.5 character times of the silent line
Address	2 characters	1 byte
Function	2 characters	1 byte
Data address	4 characters	2 bytes
Data value	4 characters	2 bytes
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

The format of a response to a force multiple coils or a preset multiple registers function query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character time of the silent line
Address	2 characters	1 byte
Function	2 characters	1 byte
Data address	4 characters	2 bytes
Quantity of points	4 characters	2 bytes
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

The format of a response to a diagnostics function query is an echo of the query itself. If the request is directed to a counter, however, the slave will return the counter's value in the data field.

1.4. Exception responses

The format of an exception response to a master's query is as follows:

Mode	ASCII	RTU
Start character	:	3.5 character time of the silent line
Address	2 characters (echo of master's query)	1 byte (echo of master's query)
Function	2 characters (echo of master's query with MSB set)	1 byte (echo of master's query with MSB set)
Exception code	2 characters	1 byte
Error check field	LRC 2 characters	CRC 2 bytes
End characters	CRLF (0x0D 0x0A)	3.5 character times of the silent line

The application program in the master is responsible for handling exception responses. Typical processes include successive attempts to send a query, sending diagnostic messages to the slave, and notifying the operators.

2. Modbus REX 521 profile

The Modbus protocol (ASCII or RTU mode) is selected via a HMI (human-machine interface) and can be used through two rear connection options:

- RS-485 interface (twisted-pair) or
- fibre-optic interface.

The rear connection option and the Modbus line settings (parity, CRC byte order, and baud rate) can be adjusted either via a HMI or the SPA bus.

2.1. Supported application functions

The implementation of the Modbus protocol in the REX 521 supports the following functions:

Table 2.1.-1 Supported application functions

Function code (HEX)	Function description
01	Read coil status Reads the status of discrete outputs.
02	Read digital input status Reads the status of discrete inputs.
03	Read holding registers Reads the contents of output registers.
04	Read input registers Reads the contents of input registers.
05	Force single coil Sets the status of a discrete output.
06	Preset single register Sets the value of a holding register.
08	Diagnostics Checks the communication system between master and slave.
0B	Get comm event counters Returns amount of successful read/write operations on data points.
0F	Force multiple coils Sets the status of multiple discrete outputs.
10	Preset multiple registers Sets the value of multiple holding registers.
17	Read/write holding registers Exchanges holding registers in one query.

2.2. Supported diagnostic subfunctions

The implementation of the Modbus protocol in REX 521 supports the following subfunction codes:

Table 2.2.-1 Supported diagnostic subfunctions

Code (HEX)	Name	Subfunction description
00	Return query data	The data in query data field is returned (looped back) in response. The entire response is identical to the query.
01	Restart communication option	The peripheral port of the slave is initialized and restarted, and the communication event counters are cleared. Before this, a normal response is sent unless the port is in the listen only mode. If the port is in the listen only mode, no response will be sent.

Table 2.2.-1 Supported diagnostic subfunctions (Continued)

Code (HEX)	Name	Subfunction description
02	Return Diagnostic Register	The contents of the slave's diagnostic register is returned in response.
04	Force listen only mode	The slave is forced to enter the listen only mode for the Modbus communication.
10	Clear counters and diagnostic register	All the counters and the diagnostic register are cleared.
11	Return bus message count	The number of messages in the communications system detected by the slave since its restart, clear counters operation or power-up is returned in response.
12	Return bus communication error count	The number of CRC errors encountered by the slave since its restart, clear counters operation or power-up is returned in response.
13	Return bus exception error count	The number of Modbus exception responses sent by the slave since its restart, clear counters operation or power-up is returned in response.
14	Return slave message count	The number of messages addressed to the slave or broadcast which the slave has processed since its restart, clear counters operation or power-up is returned in response.
15	Return slave no response count	The number of messages addressed to the slave for which a response (neither a normal response nor an exception response) has not been sent since its restart, clear counters operation or power-up is returned in response.
16	Return slave NACK response count	The number of messages addressed to the slave (for which a NACK response has been sent) is returned in response.
18	Return bus character overrun count	The number of messages addressed to the slave for which it has not been able to send a response due to a character overrun since its last restart, clear counters operation or power-up is returned in response.

Sending other subfunction codes than the ones listed above cause an "*Illegal data value*" response.

2.3. Diagnostic counters

The Modbus protocol provides the following diagnostic counters:

Table 2.3.-1 Diagnostic counters

Name	Meaning
Bus message count	The number of messages in the communications system detected by the slave since its restart, clear counters operation or power up.
Bus communication error count	The number of CRC or LRC errors encountered by the slave since its restart, clear counters operation or power up.
Bus exception error count	The number of Modbus exception responses sent by the slave 610 since its restart, clear counters operation or power up.
Slave message count	The number of messages addressed to the slave or broadcast which the slave has processed since its restart, clear counters operation or power up.
Slave no response count	The number of messages addressed to the slave for which a response (neither a normal response nor an exception response) has not been sent since its restart, clear counters operation or power up.

Table 2.3.-1 Diagnostic counters (Continued)

Name	Meaning
Bus character overrun count	The number of messages addressed to the slave for which it has not been able to send a response due to a character overrun since its restart, clear counters operation or power up.

2.4. Possible exception codes

The following exception codes may be generated by the Modbus protocol:

Table 2.4.-1 Exception codes

Code (HEX)	Name	Meaning
01	Illegal function	The slave does not support the requested function.
02	Illegal data address	The slave does not support the data address, or the number of items in the query is incorrect.
03	Illegal data value	The value in the query data field is out of range.

If an "*Illegal data value*" exception response is generated when attempting to preset multiple registers, the contents of the register to which an illegal value has been imposed and the following registers will not be changed. The registers which have already been preset will not be restored.

2.5. User-defined registers

The reading of unwanted data spends bandwidth and complicates data interpretation. For optimum efficiency in Modbus communication, the master device should therefore scan only consecutive blocks of data. For this purpose, a set of programmable user-defined registers (UDR) has been defined in the holding register area.

The first 32 holding and input registers (HR/IR), that is, HR/IR 1...32 are user-defined registers. The UDRs can be linked to any holding register using SPA parameters 504V301...504V332. However, one UDR cannot be linked to another, that is, linking cannot be nested. Each parameter contains the address of the holding register to which the UDR is linked. The address is in a decimal format.

If a UDR is linked to a non-existent holding register, the reading from the register will fail and an "Illegal address exception" response will be sent. Giving the value 0 to the link address disables the UDR. If the master reads from a disabled UDR, the UDR value 0 will be returned.

2.6. Digital inputs

As the master may not detect the changes of states of all digital signals when scanning, an additional change detect (CD) indication bit is created for every momentary indication point; see the example in Figure 2.6.-1.

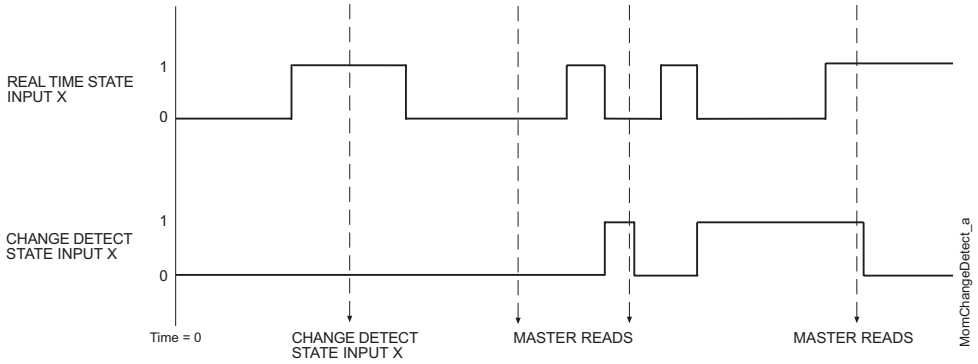

Fig. 2.6.-1 Change detection bit

If the momentary value of an indication bit has changed two or more times since the master last read it, the CD bit will be set to 1. When the CD bit has been read, it will be set to 0.

The momentary and the CD bit of a certain indication point always occur as a pair in the Modbus memory map.

2.7. Protocol parameters

The protocol and link parameters of the Modbus interface can be programmed by means of a local HMI by selecting `Configuration\Communication\Comm.settings\Modbus`.

The following parameters are available:

Table 2.7.-1 Modbus protocol parameters

Parameter	Comment	Default value
Unit address	The Modbus unit address 1…247	1
Baud Rate	Baud rates: 600/1200/ 2400/4800/9600/19200 bps	9600
Modbus Mode	Link mode: 0 = ASCII 1 = RTU	1
No of data bits	7 - ASCII Mode 8 - RTU Mode	8
No of stop bits	1…2	1
Parity	0 - No parity 1 - Odd parity 2 - Even parity	2
End of frame TO	Minimum idle time following the frame transmission to REX 521 (in milliseconds)	3
CRC Order	Order of CRC bytes in the RTU link mode 0 - Low High 1 - High Low	0

When using RTU mode, every time the parameter “Baud Rate” is reconfigured (value is changed) the software of REX recalculates the default value for the timeout parameter (“End of frame TO”) according to the standard. Therefore, to manually enforce the timeout value, the timeout parameter must always be set after configuring the “Baud Rate” parameter, never before. In this case it is recommended to set the parameter value for longer timeout than the calculated default value with the “Baud Rate” parameter in question.

Notice that the Modbus protocol should be selected for the rear port before the automatic recalculation of the time out value is done.

2.8. General guidelines on how REX 521 application data is seen on Modbus protocol

The Figure 2.8.-1 and the Table 2.8.-1 on page 16 describe how the process data in a REX 521 device is seen on the Modbus protocol.

The mapping of the function block data into the Modbus process data is done by the Modbus Protocol Object Dictionary (POD).

In the application example below, all the possible process data is present. The grey boxes show to which Modbus data category the signals belong.

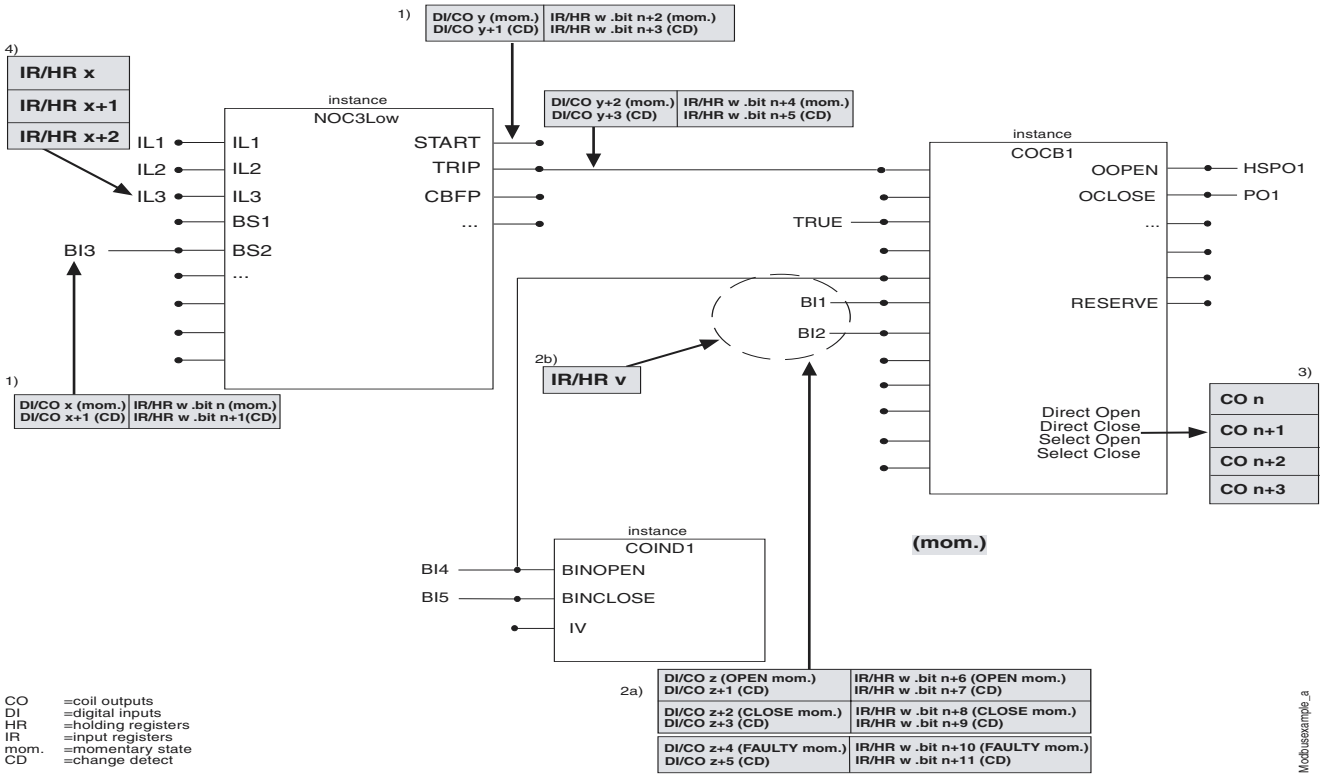

Fig. 2.8.-1 Application example

The Figure 2.8.-1 is explained in the following table:

Table 2.8.-1 Explanations to application example

No	Application data type	Explanation	Modbus data type
1)	One Bit Input	Binary input to a function block, e.g. blocking input.	Coils (0x references Digital inputs (1x references) Input Registers (3x references) Holding Registers (4x references)
1)	One Bit Output	Binary output from function block, e.g. START or TRIP signals.	Coils (0x references Digital inputs (1x references) Input Registers (3x references) Holding Registers (4x references)
2a)	Two Bit Input	Binary position data coded in two bits (OPEN, CLOSE)	Coils (0x references Digital inputs (1x references) Input Registers (3x references) Holding Registers (4x references) Coded in three bits: Bit 1: OPEN Bit 2: CLOSE Bit 3: FAULTY (validity): 1 if corresponding binary inputs of both Bit 1 and Bit 2 = 1. In case validity bit = 1, both OPEN value and CLOSE value are set to 0 in Modbus data.
2b)	Two Bit Input	In addition to 2a) the OPEN and the CLOSE bit values are also coded as least significant bits in an input and holding register. (One register per object.)	Input Registers (3x references) Holding Registers (4x references) Values: 1 = CLOSE 2 = OPEN 3 = Undefined 0 = Undefined
3)	Control output points	Outputs controlled from the Modbus master.	Coils (0x references)
4)	Measurement inputs	Measurement inputs to the function blocks	Input Registers (3x references) Holding Registers (4x references)
Not visible in the figure	Parameters, settings, etc.	Some parameters of the device and function blocks may be adjustable (look in the Modbus point list of the Modbus configuration)	Input Registers (3x references) Holding Registers (4x references) Note! Writable only to the Holding Registers.

2.9. Modbus data mapping

There are two types of monitoring data: digital indications and measurands. For convenience and efficiency, the same data can be read from different data areas. Measurands and other 16-bit values can be read either from the IR or HR (read-only) area and digital indication values from either the DI or coil (read-only) area. It is also possible to read the status of the DIs as packed 16-bit registers from both the IR and HR area.

Consequently, all the monitoring data can be read as consecutive blocks of data from the IR or HR area.

However, there are also write-only coils in the Modbus data mapping of REX 521. Those coils are marked as writable (W) in the point lists of application mappings. Naturally, this data does not occur in the DI area.

The response time for the data listed under Slowly changing and Device data categories is longer than with other data categories. If possible, it is recommended to poll Slowly changing and Device data categories less frequently than others to save the bandwidth.

The application mappings (that is, register and bit addresses) for different standard configurations of REX 521 are presented in CD-ROM *Technical Descriptions of Functions*, rev. 2.7 or later (see See “References” on page 19.)

Table 2.9.-1 Example of application mapping point list (addresses are decimal numbers)

Description	HR/IR address (.bit)	DI/Coil bit address	Writable	Value range	Scale	Unit
User defined						
MOD_BUS_REX-A, Special register 1	00001					
MOD_BUS_REX-A, Special register 2	00002					
MOD_BUS_REX-A, Special register 3	00003					
…	…					
MOD_BUS_REX-A, Special register 32	00032					
Process data						
COCB1-B, Object state	00518					
…	…					
MECU3A-C, IL1	00522			0.0...20000.0	x 10	[A]
MECU3A-C, IL2	00523			0.0...20000.0	x 10	[A]
MECU3A-C, IL3	00524			0.0...20000.0	x 10	[A]
…	…					
MEVO1A-E, Uo	00529			0...150000		[V]
DIPO-A, Input 1 state	00530.00	00512		0... 1		
DIPO-A, Input 1 state CD	00530.01	00513		0... 1		
DIPO-A, Input 2 state	00530.02	00514		0... 1		
DIPO-A, Input 2 state CD	00530.03	00515		0... 1		
…	…	…				
COCB1-B, Object state closed	00537.08	00632				
COCB1-B, Object state CD	00537.09	00633				
COCB1-B, Object state open	00537.10	00634				
COCB1-B, Object state CD	00537.11	00635				
COCB1-B, Object state faulty	00537.12	00636				
COCB1-B, Object state CD	00537.13	00637				
…	…	…				
NOC3LOW-D, Output START	00540.01	00673		0...1 [0 = Not active; 1 = Active]		
NOC3LOW-D, Output START CD	00540.02	00674		0...1 [0 = Not active; 1 = Active]		
Slowly changing						
ERHA-B, Software reset	00541		W	0...1 [0 = 0; 1=Reset;]		
…	…					

Table 2.9.-1 Example of application mapping point list (addresses are decimal numbers) (Continued)

Description	HR/IR address (.bit)	DI/Coil bit address	Writable	Value range	Scale	Unit
DOHA-A, Test mode		00676		0...1 [0=No test; 1=Testing]		
Control						
COCB1-B, Direct open		08192	W	1=Direct open		
COCB1-B, Direct close		08193	W	1=Direct close		
…	…					
Device data						
MOD_BUS_REX-A, Device clock register	04096		W			
MOD_BUS_REX-A, Device clock register	04097		W			
MOD_BUS_REX-A, Device clock register	04098		W			
MOD_BUS_REX-A, Device clock register	04099		W			
MOD_BUS_REX-A, Device clock register	04100		W			
MOD_BUS_REX-A, Device clock register	04101		W			
MOD_BUS_REX-A, Device clock register	04102		W			
…						

Table 2.9.-2 REX 521 device clock structure in Modbus application mappings

Address	Description	Range
4096	Year	0...65535
4097	Month	1...12
4098	Day	1...31
4099	Hour	0...23
4100	Minute	0...59
4101	Second	0...59
4102	Milliseconds	0...999

Table 2.9.-3 Other REX 521 device data in Modbus application mappings

Address	Description
4103	Config build nr, ASCII Chrs 1-2 (HI-LO order)
4104	Config build nr, ASCII Chrs 3-4 (HI-LO order)
4105	Config build nr, ASCII Chrs 5-6 (HI-LO order)
4106	Config build nr, ASCII Chrs 7-8 (HI-LO order)
4107	Config build nr, ASCII Chrs 9-10 (HI-LO order)
4108	Config name, ASCII Chrs 1-2 (HI-LO order)
4109	Config name, ASCII Chrs 3-4 (HI-LO order)
4110	Config name, ASCII Chrs 5-6 (HI-LO order)
4111	Config name, ASCII Chrs 7-8 (HI-LO order)
4112	Config name, ASCII Chrs 9-10 (HI-LO order)
4113	Config revision, ASCII Chrs 1-2 (HI-LO order)
4114	Serial Number, HI_WORD
4115	Serial Number, LO_WORD

3. References

Manuals for REX 521

• Installation Manual	1MRS 750526-MUM
• Operator’s Manual	1MRS 751107-MUM
• Technical Reference Manual, General	1MRS751108-MUM
• Technical Reference Manual, Standard Configurations	1MRS751802-MUM
• Technical Descriptions of Functions (CD-ROM)	1MRS750889-MCD

Parameter and event lists for REX 521

• Parameter List for REX 521	1MRS751999-RTI
• Event List for REX 521	1MRS752000-RTI
• General Parameters for REX 521	1MRS752156-RTI
• Interoperability List for REX 521	1MRS752157-RTI

Tool-specific manuals

• CAP505 Installation and Commissioning Manual	1MRS751273-MEN
• CAP505 Operator’s Manual	1MRS751709-MEN
• Tools for Relays and Terminals, User’s Guide	1MRS752008-MUM
• CAP 501 Installation and Commissioning Manual	1MRS751270-MEN
• CAP 501 Operator’s Manual	1MRS751271-MUM

4. Appendix A: Profile checklist

MODBUS DEVICE PROFILE DOCUMENT
Vendor Name: ABB Oy Substation Automation
Device Name: REX 521
Device Function: Slave
Modes: RTU ASCII

Supported function codes

Code (HEX)	Function	Supported
01	Read coil Status	Yes
02	Read Input Status	Yes
03	Read Holding Register	Yes
04	Read Input Registers	Yes
05	Force Single Coil	Yes
06	Preset Single Register	Yes
07	Read Exception Status	No
08	Diagnostics	Yes
0B	Fetch Comm Event Counter	Yes
0C	Fetch Comm Event Log	No
0F	Force Multiple Coils	Yes
10	Preset Multiple Registers	Yes
11	Report Slave ID	No
14	Read General Reference	No
15	Write General Reference	No
16	Mask Write 4x Register	No
17	Read/Write 4x Registers	Yes

Supported diagnostics subfunction codes

Code (HEX)	Name	Supported
00	Return Query Data	Yes
01	Restart Communication Option	Yes
02	Return Diagnostic Register	Yes
03	Change ASCII Delimiter	No
04	Force Listen Only Mode	Yes
0A	Clear Counters and Diagnostics Register	Yes
0B	Return Bus Message Count	Yes
0C	Return Bus Communication Error Count	Yes
0D	Return Bus Exception Error Count	Yes
0E	Return Slave Message Count	Yes
0F	Return Slave No Response Count	Yes
10	Return Slave NAK Count	Yes
11	Return Slave Busy Count	No
12	Return Bus Character Overrun Count	Yes
13	Return IOP Overrun Count	No
14	Clear Overrun Counter and Flag	No
15	Get / Clear Modbus Plus Statistics	No

Supported exception responses

Code (HEX)	Name	Supported
01	ILLEGAL FUNCTION	Yes
02	ILLEGAL DATA ADDRESS	Yes
03	ILLEGAL DATA VALUE	Yes
04	SLAVE DEVICE FAILURE	No
05	ACKNOWLEDGE	No
06	SLAVE DEVICE BUSY	No
07	NEGATIVE ACKNOWLEDGE	No
08	MEMORY PARITY ERROR	No

Supported data types

Name	Supported
Digital input	Yes
Coil	Yes
Input register	Yes
Holding register	Yes
General reference	No

Supported event reporting methods

Name	Supported
Momentary change detect on digital input	Yes

5. Appendix B: Abbreviations

ASCII	American Standard Code for Information Interchange
CD	Change Detect
CO	Coil
CRC	Cyclic Redundancy Check
DI	Digital Input
HMI	Human-machine interface
HR	Holding Register
IR	Input Register
LRC	Longitudinal Redundancy Check
PLC	Programmable Logic Controller
POD	Protocol Object Dictionary
RTU	Remote Terminal Unit
UDR	User-Defined Register

ABB Oy
Substation Automation
P.O. Box 699
FIN-65101 VAASA
Finland
Tel. +358 10 22 11
Fax. +358 10 224 1094
www.abb.com/substationautomation

1MRS 755017 EN 11.2003